Using *Jason* to Develop a Team of Cowboys (a preliminary design for Agent Contest 2008)

Jomi F. Hübner[†], Rafael H. Bordini[⋆], and Gauthier Picard[†]

[†] École des Mines de Saint-Étienne, France
{hubner,picard}@emse.fr

[⋆] University of Durham, UK
r.bordini@durham.ac.uk

1 Introduction

This document describes an overview of a multi-agent system formed by a team of cowboys to compete in the Multi-Agent Programming Contest 2008 (the "Cows and Herders" scenario). In the two previous contests we tested and improved ***Jason***, the interpreter for an agent programming language used to implement the MAS. ***Jason*** [2] is an agent platform based on an extension of an agent-oriented programming language called AgentSpeak(L) [5]. The language is inspired by the BDI architecture [6], hence based on notions such as goals, plans, beliefs, intentions, etc. The participation in the last contests also increases our experience both in programming agents with ***Jason*** and in using BDI concepts. In the 2006 contest, the focus was on the definition of agent's plans [1], leading to rather reactive agents. In the 2007 contest, the focus was on (declarative) goals [3], leading to more pro-active, goal-directed agents.

This year, we are motivated to continue to improve the abstraction level towards social or organisational agents, using the concepts of roles and groups. Besides the agents, we plan to also focus on their organisation. The system is therefore developed in two dimensions: agents (using declarative goals) and organisation (using groups, roles, and shared goals). Among several organisational models available, we will use the $\mathcal{M}\textsc{oise}^+$ model because it is well integrated with ***Jason*** [4]. Our objective in participating in this contest is thus twofold: (*i*) to continue to test and improve ***Jason*** and its integration with $\mathcal{M}\textsc{oise}^+$; (*ii*) evaluate the use of organisational constructs in the development of the team.

2 System Analysis and Design

Based on the description of the scenario, one notices that it is important that the cowboys work as a coordinated team. It is very difficult for a cowboy alone to herd a group of cows. We therefore adopt a strategy strongly tied to the notion of group of agents where issues such as spatial formation, membership, and coordination should be emphasised.

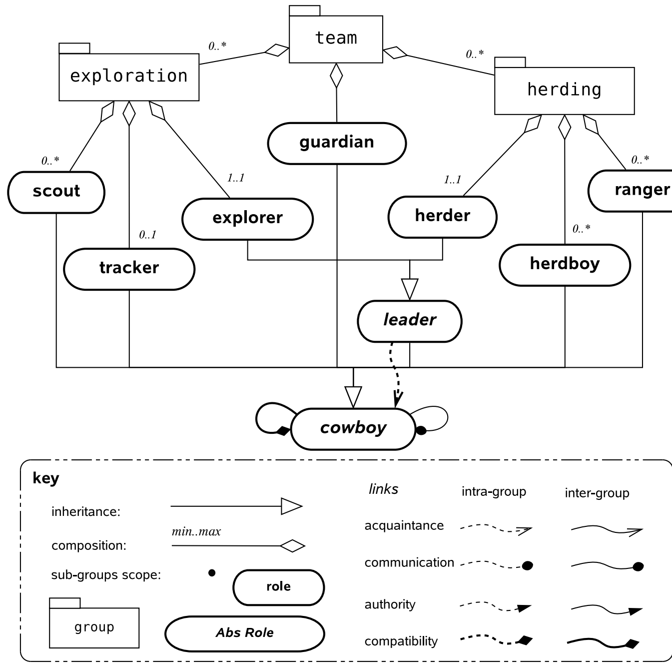

Fig. 1. The Structural Specification of the Organisation.

The team and its groups will be specified using the $\mathcal{M}$OISE$^+$ language for the organisational concepts and sequence diagrams for the *communication* protocols.[1] The overall organisational structure of the team is specified in Figure 1 using the $\mathcal{M}$OISE$^+$ notation. Our team has two subgroups: one to explore the environment seeking for cows (the *exploration group*) and another one that leads the herd toward the corral (the *herding group*). The following roles can be played in these groups:

- *guard*: guards the corral so as to keep the herded cows safe inside it;
- *explorer*: explore the environment until it detects a cow;
- *scout*: follows the explorer;
- *tracker*: once a cow is detected, tracks all cows of a cluster so as to evaluate its size;
- *herder*: herds the cows detected by explorer to reach the corral (since they move quicker than cows, they can also continue to explore around the cluster);
- *herdboy*: helps the herder to lead cows to the corral;
- *ranger*: finds the "best" path to the corral.

[1] Initially, we do not plan to use an AOSE methodology nor some tool to design the agents, as we did in previous years.

In this preliminary version of the specification we still do not have the definition of the team's shared goals and how they are related and allocated to agents (the functional specification language of $\mathcal{M}$OISE$^+$ will be used for that). Nevertheless, the general dynamics of the agents playing the above roles is described with the help of the following scenario.

- At the beginning of the simulation, the exploration group(s) are created with agents playing the explorer and scout roles. They split themselves so as to cover as wide a range as possible, without necessarily keeping each other in sight.
- As soon as an agent perceives cows, it informs the members of its exploration group. This agent plays then the tracker role to evaluate the size of the cluster, in order to evaluate it for a possible negotiation between several explorers and to decide the number of required agents to form a herding group.
- Once the cluster is evaluated, the exploration group is dissolved and a new herding group is created with as many players as necessary to lead the herd to the corral. Agents of the exploration group have preference to belong to the new group, although new members could be negotiated if necessary. The new group should decide the best agent to play the ranger role and also the herder role that will then lead the cows to the corral.
- Once the corral is reached, the ranger evaluates how safe the corral is and if necessary becomes a guard. The remaining agents form a new search group that starts exploring the environment again. If there is a guard still in place when the herding group is arriving, it becomes a member of the new search group (as soon as the arriving ranger enters its perception range).

Note that we are trying to balance when to distribute decisions (some issues are negotiated) and when to centralise them (in agents playing a leader role as the herder and the explorer). Roughly, global strategic decisions are negotiated and local (intra-group) functioning/operational decisions are taken by the leader.

Although we have some global constraints over the agents' behaviour (based on the roles they are playing), they are *autonomous* to decide how to achieve the goals assigned to them. While *coordination* and *team working* are managed my the $\mathcal{M}$OISE$^+$ tools, the *autonomy* and *pro-activeness* are facilitated by the BDI architecture of our agents implemented in ***Jason***. Regarding *communication*, we use the speech-act based communication language available in ***Jason***.

3 Software Architecture

To implement our agent team, two features of ***Jason*** were specially useful: architecture customisation and internal actions (see Figure 2). A customisation of the agent architecture is used to interface between the agent and its environment. The environment for the Agent Contest is implemented in a remote server that simulates the cattle field, sending perception to the agents and receiving requests for action execution. Therefore, when an agent attempts to perceive

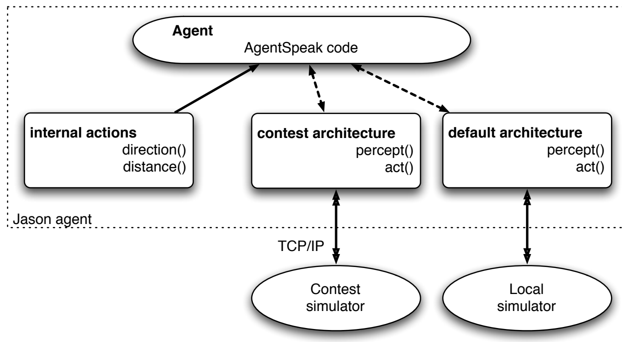

Fig. 2. Agent Architecture.

the environment, the customised architecture sends to the agent the information provided by the simulator, and when the agent chooses an action to be performed, the architecture sends the action execution request to the simulator. This architecture customisation also allow us to easily change the contest simulator to our local simulator by simply choosing another architecture; using a local simulator makes testing much easier and faster.

Although most of the agent code will be written in AgentSpeak, some parts are to be implemented in Java, in this case because we aim to use legacy code. In particular, we already had a Java implementation of the A* search algorithm, which we plan to use to find paths and calculate distances in the various scenarios of the competition. This algorithm was made accessible to the agents by means of *internal actions*. Custom architecture and internal actions are also used to the perceive the current organisation of the team and to change it, respectively.

4 Agent team strategy

1. Navigation algorithms:
   - obstacle avoiding
     As in previous teams, we will use the A algorithm to avoid obstacles.*
   - strategy for finding and herding cows
     The group formation is the base for these tasks. To find cows, we will start with previous team strategies: division of field and least visited locations. Other strategies more suitable for the wide vision area will be studied.
   - opponent blocking
     We will place a guard at the door of our corral and the ranger will (try to) lead the herding group on a path without opponents.
2. Describe the team coordination strategy (if any)
   The coordination is based on shared global goals and global plans as defined in $\mathcal{M}$OISE$^+$.

3. Does your team strategy use some distributed optimization technique w.r.t. e.g. minimizing distances walked by the agents?
   In general, no, but negotiation might be used to find out good solutions. At the individual level, A finds optimal paths.*
4. Describe and discuss the information exchanged (and shared) in the agent team.
   The more information (specially obstacles) about the scenario is available for A, the better it performs. So when an agent sees an obstacle, it broadcasts this information to all agents so that they can update their world model accordingly.*
5. Describe the communication strategy in the agent team. Can you estimate the communication complexity in your approach?
   We have not yet defined the communication protocols.
6. Did you system do some background processing? Under background processing we understand some computation which happened while agents of the team were *idle*, i.e. between sending an action message to the simulation server and receiving a perception message for the subsequent simulation step.
   No.
7. Possibly discuss additional technical details of your system like e.g. failure/crash recovery and alike.
   We plan to associate an "angel" to each agent; the angel checks if the agent is blocked/crashed and then tries to solve the problem automatically.

References

1. R. H. Bordini, J. F. Hübner, and D. M. Tralamazza. Using *jason* to implement a team of gold miners. In K. Inoue, K. Satoh, and F. Toni, editors, *Proc. of the 7th Workshop on Computational Logic in Multi-Agent Systems (CLIMA VII)*, volume 4371 of *LNCS*, pages 304–313. Springer, 2007.
2. R. H. Bordini, J. F. Hübner, and M. Wooldrige. *Programming Multi-Agent Systems in AgentSpeak using Jason.* John Wiley & Sons, 2007.
3. J. F. Hübner and R. H. Bordini. Developing a team of gold miners using *jason*. In M. Dastani, A. E. F. Seghrouchni, A. Ricci, and M. Winikoff, editors, *Proceedings of the Fifth international Workshop on Programming Multi-Agent Systems (ProMAS 07)*, volume 4908 of *LNAI*, pages 241–245. Springer, 2008.
4. J. F. Hübner, J. S. Sichman, and O. Boissier. Developing organised multi-agent systems using the MOISE+ model: Programming issues at the system and agent levels. *International Journal of Agent-Oriented Software Engineering*, 1(3/4):370–395, 2007.
5. A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. Van de Velde and J. Perram, editors, *Proceedings of the Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, number 1038 in LNAI, pages 42–55, London, 1996. Springer-Verlag.
6. A. S. Rao and M. P. Georgeff. BDI agents: from theory to practice. In V. Lesser, editor, *Proceedings of the First International Conference on MultiAgent Systems (ICMAS'95)*, pages 312–319. AAAI Pess, 1995.